

*A nuestras personas queridas*



... pleasure has probably been the main goal all along. But I hesitate to admit it, because computer scientists want to maintain their image as hard-working individuals who deserve high salaries. Sooner or later society will realise that certain kinds of hard work are in fact admirable even though they are more fun than just about anything else.

*Donald Edwin Knuth*

En aquel Imperio, el Arte de la Cartografía logró tal Perfección que el mapa de una sola Provincia ocupaba toda una Ciudad, y el mapa del Imperio toda una Provincia. Con el tiempo, esos Mapas Desmesurados no satisficieron y los Colegios de Cartógrafos levantaron un Mapa del Imperio, que tenía el tamaño del Imperio y coincidía puntualmente con él. Menos Adictas al Estudio de la Cartografía, las Generaciones Sigüientes entendieron que ese dilatado Mapa era Inútil y no sin Impiedad lo entregaron a las Inclemencias del Sol y de los Inviernos. En los desiertos del Oeste perduran despedazadas Ruinas del Mapa, habitadas por Animales y por Mendigos; en todo el País no hay otra reliquia de las Disciplinas Geográficas.

*Jorge Luis Borges*  
*Museo*

Programming is a form of art. Not only is elegance and beauty possible in programming computers, these are at the core of a good programmer's value system. Computers present to mankind the first opportunity to do what religions all over the world have failed to present: the ability to receive unambiguous answers to incantations and prayers —computers are the man-made gods who *listen*. I am, of course, talking metaphorically about instructing computers to create what we want to exist.

Like other information should be available to those who want to learn and understand, program source code is the only means for programmers to learn the art from their predecessors. It would be unthinkable for playwrights not to allow other playwrights to read their plays, only be present at theater performances where they would be barred even from taking notes. Likewise, any good author is well read, as every child who learns to write will read hundreds of times more than it writes. Programmers, however, are expected to invent the alphabet and learn to write long novels all on their own. Programming cannot grow and learn unless the next generation of programmers have access to the knowledge and information gathered by other programmers before them.

*Erik Naggum*  
*Ideas and Principles. Programming*  
<http://www.naggum.no/erik/programming.html>  
6 de Febrero de 1996
















El tangram es un juego de mesa originario de China. Consta de siete piezas con formas geométricas muy básicas; las puedes ver en el comienzo de este párrafo: dos triángulos rectángulos grandes, dos pequeños, uno mediano, un cuadrado y un paralelogramo. Los triángulos pequeños son los ladrillos básicos; juntando dos se obtiene el cuadrado, el paralelogramo o el triángulo mediano, dependiendo de cómo se haga, y juntando cuatro se obtienen los triángulos grandes.



Consiste el juego en la recreación de una figura haciendo uso de las siete piezas. Al principio del párrafo anterior se muestra la disposición que genera un cuadrado; en éste, la que genera un cisne; en el siguiente ya no se revela el secreto de cómo construir una liebre, aunque debería ser evidente.



El esfuerzo intelectual del tangram depende de qué figura se tenga que conseguir; el cuadrado es una de las más difíciles. El esfuerzo también depende de si nos ofrecen alguna pieza ya colocada: así, los problemas de completar las figuras , ,  y  para obtener un cuadrado son progresivamente más complejos. Cada ejercicio se adorna con uno de estos iconos para indicar su dificultad. Las imágenes , ,  y  marcan dónde hay que colocar alguna de las piezas que faltan; las usaremos para señalar que un ejercicio tiene pistas. Finalmente, con la figura  indicaremos que el ejercicio está resuelto. Los iconos para pistas y soluciones siempre irán acompañados del número de la página a la que recurrir.

**Código** Forzados por la exposición, muchas soluciones presentan un código fragmentado en piezas. Siempre que su reorganización y unión en forma de programa completo sea obvia, evitaremos el gasto de ofrecer este resultado en papel.

Muchas veces, en puntos intermedios de una explicación, ofreceremos código que carece de algún detalle interno. En su lugar aparecerá una frase explicativa; para resaltar que es inadmisibles como C++ correcto, se presentará así:

*⟨Una frase explicativa del código ausente⟩*

**Prolongación** En <http://aljibe.sip.ucm.es> puedes encontrar, además de muchos otros problemas, soluciones en otros lenguajes de programación y diversos recursos que no tiene sentido poner en papel, como códigos completos, datos de pruebas, etc.

Pero la iniciativa de recopilar problemas de programación y material relacionado no es nueva ni exclusivamente nuestra. Consulta los ejercicios de la Olimpiada Informática Española (<http://www.aula-ee.com/oie/probs.htm>) o los del concurso internacional de la ACM de problemas de programación (*The ACM International Collegiate Programming Contest*, <http://icpc.baylor.edu/icpc/>). Estos últimos se han recopilado en <http://acm.uva.es/problemset/> y complementado con una herramienta que comprueba soluciones.

Finalmente, no dudes en consultar esa enciclopedia, inmensa y cambiante, que Internet y el *software libre* redactan día a día para quienes disfrutan de la práctica de la programación: <http://www.gnu.org>, <http://www.opensource.org>, <http://www.sourceforge.org>, etc.

**Agradecimientos** Como puede verse, en la portada del libro figuran cinco autores; y si bien esto es verdad, no es toda la verdad: porque no podemos ignorar la valiosa contribución de Euclides, la del inventor del ajedrez, la de Julio Cortázar, el rey Wen, la tortuga de Lou Shu, la de los niños de Agra y Allahabad, la de Guido d'Arezzo, Martin Gardner, Borges, la de los campesinos rusos, los antiguos egipcios, y muchos otros cuyos nombres no podemos relacionar en su totalidad por obvias razones de espacio. Entre esos personajes ilustres, es imposible omitir a Óscar Martín Sánchez, por sus revisiones, tan profundas como acertadas, a Carmen Antón Luaces y Carmen Muñoz Serrano, por su ayuda con la bibliografía, y a David Gregorio Rodríguez, por su ayuda en la edición de las fotografías. También ha jugado un papel importante esa chispa que habita, agazapada, en el cerebro de muchos de nuestros alumnos, y que de vez en cuando salta alegremente de una neurona a otra, con regocijo de todos.
































Y a ti precisamente, que estás leyendo estas líneas en este momento, y tal vez te dispones a recorrer las páginas siguientes, te dedicamos expresamente el libro, y te pedimos indulgencia con los defectos que, sin duda, tiene. De ellos, sí somos responsables los autores de la portada y nadie más.

<b>1</b>	<b>Introducción a la programación</b>	<b>1</b>
▷	RESUMEN . . . . .	3
1.1	Contexto . . . . .	3
1.2	Comentarios . . . . .	3
1.3	Tipos básicos . . . . .	3
1.3.1	Enteros . . . . .	4
1.3.2	Caracteres . . . . .	4
1.3.3	Reales . . . . .	5
1.3.4	Valores de verdad . . . . .	6
1.3.5	Cadenas de caracteres . . . . .	6
1.3.6	Compatibilidad de tipos . . . . .	6
1.4	Identificadores . . . . .	7
1.5	Declaración de variables . . . . .	7
1.6	Definición de constantes . . . . .	8
1.7	Expresiones . . . . .	8
1.7.1	Operaciones aritméticas . . . . .	10
1.7.2	Operaciones lógicas y relacionales . . . . .	10
1.7.3	Manipulación de bits . . . . .	10
1.7.4	Expresión de selección condicional . . . . .	10
1.7.5	Conversión de tipos . . . . .	10
1.7.6	Funciones de la biblioteca estándar . . . . .	11
1.8	Instrucciones básicas . . . . .	11
1.8.1	Asignación . . . . .	11
1.8.2	Incrementos y decrementos . . . . .	12
1.8.3	Asignación con operación . . . . .	12
1.8.4	Llamada a una acción y la biblioteca estándar . . . . .	13
1.9	Cadenas de caracteres y programación orientada a objetos . . . . .	13
1.10	Rudimentos de entrada y salida . . . . .	14
1.10.1	Canales de salida . . . . .	15
1.10.2	Canales de entrada . . . . .	16
1.10.3	Ficheros o archivos . . . . .	17
1.11	Punto de arranque de un programa . . . . .	18
▷	ENUNCIADOS . . . . .	19
1.1	Espacios de color . . . . .	19  33
1.2	Ser o no ser... triángulo . . . . .	20  33
1.3	Cálculo y verificación de la letra del D.N.I. . . . .	20  34
1.4	Cuadrados perfectos . . . . .	21  34
1.5	Números triangulares . . . . .	22  34
1.6	Rectángulos . . . . .	22
1.7	Movimientos en un tablero de ajedrez . . . . .	23  36
1.8	División entera con redondeo . . . . .	23  36
1.9	División entera con préstamo . . . . .	24  37
1.10	Cálculo de la división entera y resto entre reales . . . . .	24
1.11	El beso exacto . . . . .	24  37
1.12	¿Pueden las matrices dar volteretas? . . . . .	25



1.13	Todas la funciones booleanas de dos argumentos . . . . .	25	▲
1.14	Fichas de dominó . . . . .	25	▲ 38
1.15	Expresiones relacionadas con una formación rectangular de números . . . . .	26	▲ 39
1.16	Expresiones relacionadas con una formación triangular de números . . . . .	27	▲ 40
1.17	Reflexión de los bits de una palabra de anchura fija . . . . .	27	▲ 41
1.18	Organización de torneo . . . . .	27	▲
1.19	<i>n</i> -goros . . . . .	28	▲
1.20	Cortes en una tarta . . . . .	29	▲
1.21	Número de bits a 1 . . . . .	29	▲ 42
▷	PISTAS . . . . .	31	
▷	SOLUCIONES . . . . .	33	
<b>2</b>	<b>Instrucciones estructuradas</b> . . . . .	<b>45</b>	
▷	RESUMEN . . . . .	47	
2.1	Composición secuencial . . . . .	47	
2.2	Selección condicional . . . . .	47	
2.3	Iteración . . . . .	50	
2.4	Otras instrucciones de iteración . . . . .	51	
2.4.1	Bucle <code>for</code> . . . . .	51	
2.4.2	Bucle con una ejecución . . . . .	52	
2.4.3	Interrupción de un bucle . . . . .	52	
2.5	Selección con tipos ordinales . . . . .	53	
▷	ENUNCIADOS . . . . .	55	
2.1	Signo del producto . . . . .	55	▲ 81
2.2	Ponle tú el título . . . . .	55	▲ 81
2.3	El escaso coste de la destrucción . . . . .	55	▲
2.4	Lectura de Rayuela . . . . .	56	▲ 82
2.5	¡Reflejos! . . . . .	56	▲
2.6	Código de rotación . . . . .	57	▲
2.7	Juego de adivinación . . . . .	58	▲
2.8	Evaluación de polinomios . . . . .	58	▲
2.9	Reorganiza las líneas de un texto . . . . .	59	▲
2.10	Representación gráfica de funciones . . . . .	59	▲ 82
2.11	Interés cambiante . . . . .	60	▲
2.12	De cómo quitar los comentarios de un programa . . . . .	60	▲
2.13	Un dibujo de Escher . . . . .	60	▲
2.14	Espectro natural . . . . .	61	▲
2.15	Los cubos de Nicómaco . . . . .	62	▲ 84
2.16	Un bonito triángulo . . . . .	62	▲ 85
2.17	Varianza . . . . .	63	▲ 85
2.18	ISBN de libros . . . . .	63	▲
2.19	Los hexagramas del <i>yin</i> y el <i>yang</i> . . . . .	64	▲
2.20	Suma marciana . . . . .	64	▲ 86
2.21	El número y sus representaciones . . . . .	65	▲
2.22	Códigos de Peter Elias . . . . .	67	▲
2.23	Dibujos con asteriscos . . . . .	68	▲ 88
2.24	Aproximaciones al número $\pi$ . . . . .	69	▲
2.25	Formas de determinar la pertenencia de un punto a un polígono . . . . .	71	▲

2.26	Área encerrada por unos movimientos simples . . . . .	71	▲	90
2.27	$n$ -goros . . . . .	72	▲	
2.28	De notación polaca a código ensamblador . . . . .	73	▲	91
2.29	Raíces y logaritmos . . . . .	74	▲	92
2.30	Parte entera de logaritmo . . . . .	74	▲	
2.31	El principito . . . . .	74	▲	
2.32	Suma que te suma . . . . .	75	▲	94
▷	PISTAS . . . . .	77		
▷	SOLUCIONES . . . . .	81		
<b>3</b>	<b>Subprogramas</b>			<b>97</b>
▷	RESUMEN . . . . .	99		
3.1	Funciones . . . . .	99		
3.2	Acciones o procedimientos . . . . .	101		
3.3	Elementos avanzados . . . . .	101		
3.3.1	Referencias constantes . . . . .	101		
3.3.2	Sobrecarga . . . . .	102		
3.3.3	Definición de operadores . . . . .	102		
▷	ENUNCIADOS . . . . .	103		
3.1	Incierta igualdad de números reales . . . . .	103	▲	131
3.2	Triángulos rectángulos enteros . . . . .	103	▲	
3.3	Ponle tú el título . . . . .	103	▲	131
3.4	Descomposición de un número . . . . .	104	▲	
3.5	Palíndromos . . . . .	104	▲	
3.6	Cuentaletas . . . . .	105	▲	131
3.7	Movimiento de una partícula dentro de una superficie . . . . .	106	▲	
3.8	Generación de primos . . . . .	107	▲	
3.9	Conjetura de Goldbach . . . . .	107	▲	
3.10	El factorial en la sociedad del futuro . . . . .	108	▲	
3.11	Sucesión bicicleta . . . . .	109	▲	134
3.12	Cotejo de $n$ -gramas . . . . .	109	▲	
3.13	Simulación de variables aleatorias . . . . .	110	▲	
3.14	Aproximación hacia $\pi$ con dardos . . . . .	114	▲	
3.15	Conjetura para la formación de palíndromos . . . . .	114	▲	
3.16	Juegos perdedores ganan . . . . .	115	▲	
3.17	La tabla de Galton . . . . .	116	▲	
3.18	El retrato robot . . . . .	116	▲	
3.19	Número de ceros en que termina un factorial . . . . .	118	▲	135
3.20	Representación de un número con palabras . . . . .	118	▲	
3.21	¿Cuál es el mejor orden para recibir los datos de un polinomio? . . . . .	119	▲	136
3.22	Calificación de oído . . . . .	120	▲	
3.23	Los cuadrados abominables y cáncer . . . . .	120	▲	
3.24	Codificaciones de plantas con cadenas . . . . .	121	▲	138
3.25	Triángulos anidados . . . . .	122	▲	143
3.26	Cálculo puntual de la matriz de mediotono de Judice-Jarvis-Ninke . . . . .	123	▲	144
3.27	Dibujo de árboles mediante fractales . . . . .	124	▲	146
3.28	Dragones y teselas . . . . .	126	▲	148
▷	PISTAS . . . . .	129		

▷ SOLUCIONES . . . . .	131
<b>4 Definición de tipos</b>	<b>151</b>
▷ RESUMEN . . . . .	153
4.1 Arrays . . . . .	153
4.2 De cómo nombrar tipos . . . . .	154
4.3 Registros . . . . .	155
4.4 Enumeraciones . . . . .	157
▷ ENUNCIADOS . . . . .	159
4.1 Ponle tú el título . . . . .	159  193
4.2 Yahoos . . . . .	159  193
4.3 Regla de Ruffini . . . . .	161  195
4.4 Un pequeño sistema formal . . . . .	161  195
4.5 Tratamiento de matrices como vectores . . . . .	162  197
4.6 Centro de un vector . . . . .	163  197
4.7 Un solitario . . . . .	163  197
4.8 Descomposición en sumas . . . . .	164  197
4.9 El juego del <i>master mind</i> . . . . .	165  197
4.10 AnimASCIIón . . . . .	166  201
4.11 Criterios de divisibilidad . . . . .	167  201
4.12 Otra variación sobre los números primos . . . . .	169  201
4.13 Un par de juegos con dados rodantes . . . . .	169  201
4.14 El juego de <i>sumar quince</i> . . . . .	171  201
4.15 Movimiento planetario . . . . .	172  203
4.16 Códigos para la corrección de errores . . . . .	173  203
4.17 Ajuste de imagen . . . . .	176  208
4.18 Diferencias finitas . . . . .	177  208
4.19 Búsqueda de la persona famosa en una reunión . . . . .	180  208
4.20 El efecto dominó . . . . .	180  208
4.21 Código de sustitución polialfabético . . . . .	181  208
4.22 Triángulo de Pascal . . . . .	184  211
4.23 El solitario de los <i>quince</i> . . . . .	186  211
4.24 Segmentador de oraciones . . . . .	187  214
▷ PISTAS . . . . .	189
▷ SOLUCIONES . . . . .	193
<b>5 Miscelánea de tipos</b>	<b>219</b>
▷ RESUMEN . . . . .	221
5.1 Las uniones son registros variantes . . . . .	221
5.2 Orden superior . . . . .	223
▷ ENUNCIADOS . . . . .	225
5.1 Lógica e incertidumbre . . . . .	225  257
5.2 El juego del ahorcado . . . . .	225  258
5.3 Guerra de las galaxias . . . . .	226  262
5.4 Ceros de una función . . . . .	227  262
5.5 Derivadas y desarrollos en serie . . . . .	228  265
5.6 Un sistema electoral utópico . . . . .	228  265
5.7 Sobre el juego del ajedrez . . . . .	229  265

5.8	Un traductor de Morse . . . . .	230	▲	266
5.9	Crucigramas . . . . .	231	▲	
5.10	Parchís . . . . .	233	▲	
5.11	El juego de la vida . . . . .	233	▲	
5.12	<i>Lights Out</i> . . . . .	235	▲	
5.13	Pequeña teoría de la música . . . . .	237	▲	
5.14	Números de Eudoxus . . . . .	243	▲	268
5.15	Sopas de letras . . . . .	243	▲	
5.16	Códigos lineales . . . . .	244	▲	
5.17	Códigos de trasposición utilizando rejillas . . . . .	248	▲	273
5.18	Un laberinto . . . . .	250	▲	
5.19	Implementación de números grandes . . . . .	251	▲	279
▷	PISTAS . . . . .	253		
▷	SOLUCIONES . . . . .	257		
<b>6</b>	<b>Memoria dinámica</b>	<b>291</b>		
▷	RESUMEN . . . . .	293		
6.1	Conceptos básicos . . . . .	293		
6.2	Memoria dinámica y registros . . . . .	294		
6.3	Punteros y arrays . . . . .	297		
6.4	Manejo de arrays . . . . .	297		
▷	ENUNCIADOS . . . . .	301		
6.1	Cálculo de la matriz completa de mediotono de Judice-Jarvis-Ninke . . . . .	301	▲	327
6.2	Polígonos . . . . .	301	▲	
6.3	Listas conjugadas . . . . .	301	▲	
6.4	Tiempo de conexión a una máquina . . . . .	302	▲	
6.5	Reglas golombinas . . . . .	302	▲	330
6.6	Solitario búlgaro . . . . .	303	▲	333
6.7	Una listilla con listas . . . . .	303	▲	
6.8	De cómo podar setos . . . . .	305	▲	
6.9	Cuadrados mágicos . . . . .	305	▲	338
6.10	Implementación de polinomios . . . . .	309	▲	345
6.11	Máximos y mínimos . . . . .	309	▲	352
6.12	Caminos en la red de metro . . . . .	310	▲	
6.13	El planificador que no planifique un mal planificador será . . . . .	311	▲	
6.14	Simulación de una cola múltiple . . . . .	312	▲	
6.15	Un diccionario electrónico bilingüe . . . . .	314	▲	
6.16	El sinónimo absurdo . . . . .	318	▲	
6.17	La foto de un árbol . . . . .	318	▲	
6.18	Baldosas de jardín . . . . .	321	▲	355
▷	PISTAS . . . . .	323		
▷	SOLUCIONES . . . . .	327		

- [AF00] MARCELO ALONSO Y EDWARD J. FINN. *Física*. Addison-Wesley (2000).
- [AH01] JÖRG ARDNT Y CHRISTOPH HAENEL. *Pi Unleashed*. Springer (2001).
- [AHU98] A. V. AHO, J. E. HOPCROFT Y J. D. ULLMAN. *Estructura de datos y algoritmos*. Addison-Wesley (1998).
- [AR95] OWEN ASTRACHAN Y DAVID REED. “The Applied Apprenticeship Approach to CS1”. *SIGCSE Bulletin* **27**(1), 1–5 (1995).
- [Ba102] PHILIP BALL. *Bright Earth: Art and the Invention of Color*. Farrar, Straus & Giroux (2002).
- [BB96] GILLES BRASSARD Y PAUL BRATLEY. *Fundamentals of algorithmics*. Prentice-Hall (1996).
- [BB00] GILLES BRASSARD Y PAUL BRATLEY. *Fundamentos de algoritmia*. Prentice-Hall (2000).
- [Bec82] PETR BECKMANN. *{A history of}  $\pi$* . Golem Press (1982).
- [Ben84] JOHN BENTLEY. “Programming Pearls”. *Communications of the ACM* (enero 1984).
- [BH87] R. S. BIRD Y J. HUGHES. “The Alpha-Beta Algorithm: An Exercise in Program Transformation”. En *Information Processing Letters*, volumen 24, páginas 53–57. North-Holland (1987). International Summer School on Constructive Methods in Computer Science. Marktoberdorf, Germany.
- [BHvW00] MARK BRULS, KEES HUIZING Y JARKE J. VAN WIJK. “Squarified Treemaps”. En *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization*, páginas 33–42 (2000).
- [Bir86] R. BIRD. “Tabulation techniques for recursive programs”. *ACM Computing Surveys* **12**(4), 403–417 (diciembre 1986).
- [Cha99] JEAN-LUC CHABERT, editor. *A History of Algorithms: From the Pebble to the Microchip*. Springer (1999).
- [Dew85a] A. K. DEWDNEY. “Cinco piezas sencillas para bucle y generador de números aleatorios”. *Investigación y Ciencia* **105**, 94–99 (junio 1985).
- [Dew85b] A. K. DEWDNEY. “Yin y Yang: recurrencia o iteración, la torre de Hanoi y las argollas chinas”. *Investigación y Ciencia* **100**, 102–107 (enero 1985).
- [Dew86] A. K. DEWDNEY. “Búsqueda de una regla invisible que ayude a los radioastrónomos en la medición de nuestro planeta”. *Investigación y Ciencia* **113**, 104–108 (febrero 1986).
- [dG84] MIGUEL DE GUZMÁN. “Juegos matemáticos en la enseñanza”. En *Actas de las IV Jornadas sobre Aprendizaje y Enseñanza de las Matemáticas*, Santa Cruz de Tenerife (septiembre 1984). Sociedad Canaria de Profesores de Matemáticas Isaac Newton. Accesible en <http://www.mat.ucm.es/deptos/am/guzman/juemat/juemat.htm>.
- [Dic45] L. R. DICE. “Measures of the amount of ecologic association between species”. *Ecology* **26**, 297–302 (1945).

- [Dox00] APOSTOLOS DOXIADIS. *El tío Petros y la conjetura de Goldbach*. Ediciones B (2000).
- [dP63] GIOVANNI BATTISTA DELLA PORTA. *De Furtivis Literarum Notis*. Libri Naples (1563).
- [dPC00] DIONISIO DE PEDRO CURSÁ. *Teoría completa de la música*. Real Musical (2000). dos volúmenes.
- [dvOS00] MARK DE BERG, MARC VAN KREVELD, MARK OVERMARS Y OTFRIED SCHWARZKOPF. *Computational Geometry: Algorithms and Applications*. Springer Verlag (mayo 2000).
- [Eli75] PETER ELIAS. “Universal Codeword Sets and Representations of the Integers”. *IEEE Transactions on Information Theory* **21**(2), 194–203 (marzo 1975).
- [Eli93] NORBERT ELIAS. *El proceso de la civilización: investigaciones sociogenéticas y psicogenéticas*. Fondo de Cultura Económica (1993).
- [Enz01] HANS MAGNUS ENZENSBERGEN. *El diablo de los números: un libro para todos aquéllos que temen a las matemáticas*. Siruela (2001).
- [Euc91] EUCLIDES. *Elementos. Libros I-IV*. Gredos (1991).
- [Euc94] EUCLIDES. *Elementos. Libros V-IX*. Gredos (1994).
- [Euc96] EUCLIDES. *Elementos. Libros X-XIII*. Gredos (1996).
- [Fen96] PETER FENWICK. “Punctured Elias codes for variable-length coding of the integers”. Informe técnico, University of Auckland (1996).
- [Fey98] RICHARD P. FEYNMAN. *Física*. Addison-Wesley (1998).
- [Fri98] JEFFREY E. F. FRIEDL. *Mastering Regular Expressions. Powerful Techniques for Perl and other Tools*. O’Reilly (1998).
- [FvDFH97] JAMES D. FOLEY, ANDRIES VAN DAM, STEVEN K. FEINER Y JOHN F. HUGHES. *Computer Graphics. Principles and Practice*. Addison-Wesley (1997).
- [Gar83] MARTIN GARDNER. “Tareas que es forzoso concluir por mucho que se quiera evitarlo.”. *Investigación y Ciencia* **85**, 100–105 (octubre 1983).
- [Gar86] MARTIN GARDNER. *Festival mágico-matemático*. Alianza (1986).
- [Gar87] MARTIN GARDNER. *Carnaval matemático*. Alianza Editorial, S. A (1987).
- [Gar88] MARTIN GARDNER. *Ruedas, vida y otras diversiones matemáticas*. Labor (1988).
- [Gar94] MARTIN GARDNER. *Nuevos pasatiempos matemáticos*. Alianza (1994).
- [Gar95] MARTIN GARDNER. *¡Ajá! Inspiración ¡ajá!*. Labor (1995).
- [Góm99] VÍCTOR GÓMEZ PIN. *La tentación pitagórica: ambición filosófica y anclaje matemático*. Síntesis (1999).
- [Gol91] D. GOLDBERG. “What every computer scientist should know about floating-point arithmetic”. *ACM Computing Surveys* **23**(1), 5–48 (1991).

- [Gru84] FRED GRUENBERGER. “De cómo manejar números de miles de cifras y de por qué nos es necesario”. *Investigación y Ciencia* **93**, 110–115 (junio 1984).
- [Gui75] GENEVIÈVE GUITEL. *Histoire Comparée des Numerations Écrites*. Flammarion (1975).
- [Gut00] RON GUTMAN. “Exploiting 64-bit parallelism”. *Dr. Dobb’s Journal* (septiembre 2000).
- [Hei00] ERNST A. HEINZ. *Scalable Search in Computer Chess. Algorithmic Enhancements and Experiments at High Search Depths*. Computational Intelligence. Vieweg Verlag (2000).
- [Hil99] RAYMOND HILL. *A first course in coding theory*. Oxford Clarendon Press (1999).
- [HLL<sup>+</sup>92] D. G. HOFFMAN, D. A. LEONARD, C. C. LINDNER, K. T. PHELPS, C. A. RODGER Y J. R. WALL. *Coding theory: the essentials*. Marcel Dekker (1992).
- [Hof99] DOUGLAS R. HOFSTADTER. *Gödel, Escher, Bach, an Eternal Golden Braid*. Basic Books (1999).
- [Hof01] DOUGLAS R. HOFSTADTER. *Gödel, Escher, Bach, un Eterno y Grácil Bucle*. Tusquets (2001).
- [Ifra01] GEORGES IFRAH. *Historia universal de las cifras: la inteligencia de la humanidad contada por los números y el cálculo*. Espasa Calpe (2001).
- [JJN74] J. N. JUDICE, J. F. JARVIS Y W. NINKE. “Using Ordered Dither to Display Continuous Tone Pictures”. En *Proceedings of the Society for Information Display*, páginas 161–169 (1974).
- [JS91] BEN JOHNSON Y BEN SHNEIDERMAN. “Treemaps: a space-filling approach to the visualization of hierarchical information structures”. En *Proceedings of the 2nd International IEEE Visualization Conference*, páginas 284–291 (October 1991).
- [KL93] K. KOYAMA Y T. W. LAI. “An optimal mastermind strategy”. *Journal of Recreational Mathematics* **25**, 251–256 (1993).
- [Knu77] DONALD K. KNUTH. “The computer as master mind”. *Journal of Recreational Mathematics* **9**, 1–6 (1976–77).
- [Knu95] DONALD E. KNUTH. *El arte de programar ordenadores*, volumen 2, Algoritmos semi-numéricos. Reverté (1995).
- [Man89] UDI MANBER. *Introduction to Algorithms: a Creative Approach*. Addison-Wesley (1989).
- [Man97] BENOÎT MANDELBROT. *La geometría fractal de la naturaleza*. Tusquets (1997).
- [Mar00] LUISA MARQUÉS, editor. *Fotografiando las matemáticas*. Carroggio (2000).
- [Mor84] B. J. T. MORGAN. *Elements of simulation*. Chapman and Hall (1984).
- [MP01] ÓSCAR MARTÍN SÁNCHEZ Y CRISTÓBAL PAREJA FLORES. “Two reflected analyses of Lights Out”. *Mathematics Magazine* **74**(4), 295–304 (2001).
- [MvV00] ALFRED J. MENEZES, PAUL C. VAN OORSCHOT Y SCOTT A. VANSTONE. *Handbook of applied cryptography*. CRC Press (2000).

- [O'R01] JOSEPH O'ROURKE. *Computational Geometry in C*. Cambridge University Press (febrero 2001).
- [Par01] JUAN PARRONDO. "Perder + perder = ganar. Juegos de azar paradójicos". *Investigación y Ciencia* **298** (julio 2001).
- [Pas65] BLAISE PASCAL. Des caractères de divisibilité des nombres de déduits de la somme de leurs chiffres, oeuvres complètes (1665).
- [Pas01] MICHEL PASTOUREAU. *Blue: The History of a Color*. Princeton University Press (2001).
- [Per88] YA. I. PERELMAN. *Fun with Maths and Physics*. Mir Publishers (1988).
- [POAR97] CRISTÓBAL PAREJA FLORES, MANUEL OJEDA ACIEGO, ÁNGEL LUIS ANDEYRO QUESADA Y CARLOS ROSSI JIMÉNEZ. *Desarrollo de algoritmos y técnicas de programación en Pascal*. Ra-Ma (1997).
- [Pom83] CARL POMERANCE. "A la búsqueda de los números primos". *Investigación y Ciencia* **77**, 80–99 (febrero 1983).
- [PR86] HEINZ-OTTO PEITGEN Y PETER H. RICHTER. *The Beauty of Fractals: Images of Complex Dynamical Systems*. Springer Verlag (1986).
- [PS91] FRANCO P. PREPARATA Y MICHAEL IAN SHAMOS. *Computational Geometry: An Introduction*. Springer Verlag (enero 1991).
- [PV87] L. PARDO Y T. VALDÉS. *Simulación. Aplicaciones prácticas en la empresa*. Díaz de Santos, S. A. (1987).
- [Rim93] STEVE RIMMER. *Bit-Mapped Graphics*. Windcrest/MacGraw-Hill (1993).
- [Sed89] ROBERT SEDGEWICK. *Algorithms*. Addison-Wesley (1989).
- [Shn92] BEN SHNEIDERMAN. "Tree Visualization with Tree-Maps: a 2-d Space-Filling Approach". *ACM Transactions on Graphics* **11**(1), 92–99 (1992).
- [Sue92] CAYO SUETONIO TRANQUILO. *Vidas de los doce césares*. Gredos (1992).
- [Tay15] BROOK TAYLOR. *Methodus incrementorum directa et inversa*. Londini: Impensis Gulielmi Innys (1715).
- [Tho00] JAMES R. THOMPSON. *Simulation: a Modeler's Approach*. John Wiley & Sons (2000).
- [vWvdW99] JARKE J. VAN WIJK Y HUUB VAN DE WETERING. "Cushion Treemaps: Visualization of Hierarchical Information". En *Proceedings of the 1999 IEEE Symposium on Information Visualization* (1999).
- [Wol96] STEPHEN WOLFRAM. *Cellular Automata an Complexity: collected papers*. Reading, MA: Addison-Wesley (1996).
- [Yak77] S. J. YAKOWITZ. *Computational probability and simulation*. Addison-Wesley Publishing Company (1977).







Trabajé el aire,  
se lo entregué al viento:  
voló, se deshizo,  
se volvió silencio.

Por el ancho mar,  
por los altos cielos,  
trabajé la nada,  
realicé el esfuerzo,  
perforé la luz,  
ahondé el misterio.

Para nada, ahora,  
para nada, luego:  
humo son mis obras,  
ceniza mis hechos.

...y mi corazón  
que se queda en ellos.

Ángel González